CPSC 2150 A3

Jay Seung Yeon Lee
100 357 736

Ex 1 ~ Ex 2

CPSC 2150 A3

Jay Seung yeon Lee
100 357 736

Ex 1.

$8n^2+7n$ is $O(n^2)$

$\rightarrow |f(n)| \leq |C \cdot g(n)|$ for All $n \geq 10$ $\quad n \geq N$

$\rightarrow 8(10)^2+7(10) \leq O(C \cdot 10^2)$

$\rightarrow 8(10^2)+7(10) \leq O(9 \cdot 10^2)$

$8n^2+7n$ is $O(n^2)$ when C is 9, $N \geq 10$

Ex2.

Simplify $n \cdot O(7n^2+2)$

$f(n) = n, \quad O(g(n)) = O(7n^2+2)$

$f(n) \cdot O(n) = O(f(n) \cdot O(n))$

$n \cdot O(7n^2+2) = O(7n^3+2n)$

To Simplify,

$|f(n)| \leq |C \cdot g(n)|$ for All $n \geq N$

$O(7n^3+2n) = O(n^3)$ when $C = 8$, $N \geq 10$

EX 3 ~ EX4

Ex3

Simplify $O(55n + 2\log n + 9n)$

$g(n) = 55n$
$f(n) = 2\log_n + 9n$

If $f(n) \in O(g(n))$, then $O(|f(n)| + |g(n)|) = O(|g(n)|)$

$$O(\underbrace{55n}_{g(n)} + \underbrace{2\log_n + 9n)}_{f(n)}) = O(55n) = O(n)$$

$$O(55n + 2\log n + 9n) = O(n)$$

Ex4.

Step 1.

~~4tt~~

If $f(n) \in O(g(n))$ then
$O(|f(n)| + |g(n)|) = O(|g(n)|)$

$O(g(n)) = O(n)$, $f(n) = \log_2 n$

$O(n \log_2 n) = O(n)$

Step1 = $O(n)$.

$O(f(n)) + O(g(n)) = O(|f(n)| + |g(n)|$

Step2 = $O(n^2)$
Step3 = $O(n^2)$
Step4 = $O(n)$,

meaning from Step 1 to Step4,
It will look like the following:

$$O(n) + O(n^2) + O(n^2) + O(n)$$
$$= O(2n^2 + 2n)$$

Ex4(simplification) ~ Ex5

To Simplify further,

$O(2n^2+2n) = O(n^2)$ when $n \geq 10$, $C=3$

$\underline{Ex4 = O(n^2)}$

Ex.5

Step1.= $O(n \log_2 n)$, To Simplify this step,

$g(n) = O(n)$, $f(n) = \log_2 n$

$|f(n)| \in |O(g(n))|$, therefore $O(|f(n)|+|g(n)|) = O(g(n))$

Step1 $= O(n)$

Step2:

Since this ~~for loop~~ for loop iterates n times,
Step 2a and Step 2b is equivalent to
2a: $nO(n)$ and 2b: $nO(1)$.
which can be simplified to $O(n^2)$ and $O(n)$ respectively.

Step2 $= O(n^2+n)$ or $O(n^2)$ using the property used in Step 1.

Step3 $= O(n)$

$O(f(n)) + O(g(n)) = O(g(n)) \; || \; O(|f(n)|+|g(n)|)$

$O(n) + O(n^2) + O(n) = O(n^2+2n)$

$O(n^2+2n) = O(n^2)$ when $n \geq 10$, $C=2$

$\underline{Ex5 = O(n^2)}$

Ex6 ~ Ex 7

Ex6.

Step 1 = $n+1$

Step 2 =

```
for( i=1, i<=n, i++)
        (i+1) +2
```

Since the for loop iterates n times, step 2 is equal to :

$$\frac{n(n+1)}{2} + n + 2n$$

Step 3 = $3n+9$

$$n+1+\frac{n(n+1)}{2}+n+2n+3n+9$$

$$= \frac{n(n+1)}{2}+7n+10$$

$$\text{Ex}6 = \frac{n(n+1)}{2} + 7n + 10$$

Ex.7

$$C_n = 2C_{n-1} + 2C_{n-2} + 2C_{n-3}, \quad \text{when } C_1 = 1,\ C_2 = 1,\ C_3 = 1$$

$$C_4 = 2C_3 + 2C_2 + 2C_1$$

$$= 2(1) + 2(1) + 2(1)$$

$$= 6, \quad C_4 = 6$$

$$C_5 = 2C_4 + 2C_3 + 2C_2$$

$$= 2(6) + 2(1) + 2(1)$$

$$= 14, \quad C_5 = 14$$

$$\boxed{\begin{array}{l} C_4 = 6 \\ C_5 = 14 \end{array}}$$

Ex 8

Ex 8: $C_n = C_{n-1} + 2n + 1$, when $C_1 = 1$

In general, $C_n = 1 + 2^{2(1)} + 4^{2(2)} \ldots + 2n$

Can be written as:

$\cancel{C_1} = 1$ — All the previous C_n gets cancelled out.

$\cancel{C_2} = \cancel{C_1} + 2\cdot 2$

$\cancel{C_3} = \cancel{C_2} + 2\cdot 3$ — which can be written again as:

$C_4 = \cancel{C_3} + 2\cdot 4$ — $1 + 2\cdot 2 + 2\cdot 3 + 2\cdot 4 \cdots 2n$

Which can then be written again as:

$2(1+2+3+4+\ldots+n) - 2 + 1$

or

$2(1+2+3+\ldots n) - 1 = 2\left(\frac{n(n+1)}{2}\right) - 1$

but since there was a trailing +1 per iteration including the first iteration we should add $n-1$.

Which makes it $2\left(\frac{n(n+1)}{2}\right) + n - 2$.

for 3 it would be $2\left(\frac{3(3+1)}{2}\right) + 3 - 2$, which is 13.

$$\boxed{C_n = 2\left(\frac{n(n+1)}{2}\right) + n - 2, \quad C_3 = 13}$$

EX 9

E

J

EX 9.

line A[i-1] += A[i] iterates n-1 times for n times (including when n==1 ~~when~~ since ~~n~~ when n=1, -1 applies and result in 0 which is true.)

This can be written as follows:

$n-1 + n_{-1}-1 + n_{-2}-1 + n_{-3}-1 \;...\; 1(\text{last } n)-1$

Last n (which will be 1) can be omitted, but since it can be cancelled out with the -1 that follows, will ~~be~~ leave it there for easier simplification.

Now, above equation can be re-written as follows:

$\frac{n(n+1)}{2} - n$ because there are n number of -1's and including the last n's, there are n number of n and n^{th} iterations.

Therefore, closed form of this equation is:

$C_n = C_{n-1} + \boxed{\left(\frac{n^2}{2} + \frac{n}{2} - n\right)}$ with initial value of $C_1 = 0$,

$C_3 =$

In Big O, this can be written as:

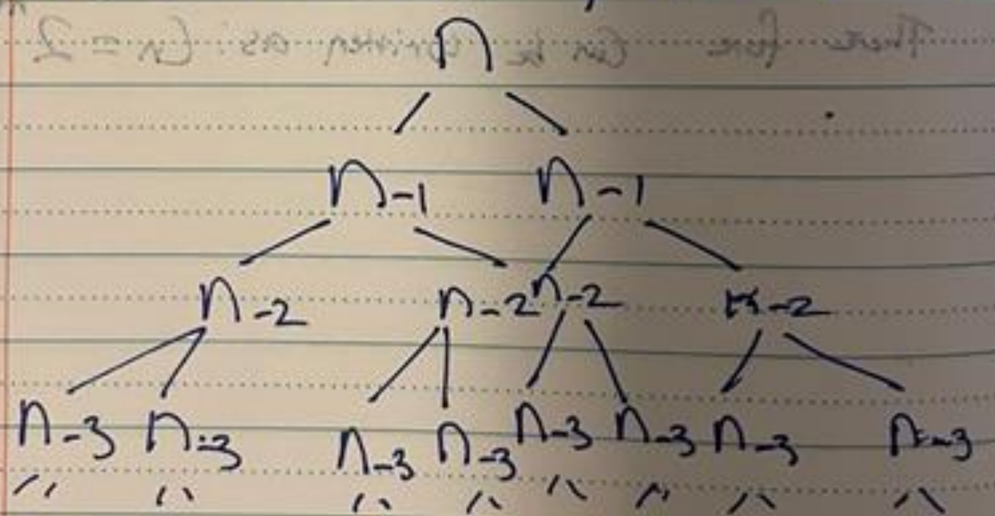
$\frac{n^2}{2} + \frac{n}{2} - n \approx \frac{n^2}{2} = \boxed{O(n^2)}$

Ex10 part 1

EX. 10

Following Code is a recursive function that calls itself twice until n==0. And further more prints out another set of prints when n==0.

When there is no Print out where n==0, Code duplicates and iterates separately. like Follows:



Closed form Should look like

$C_n = 2C_{n-1} + 1$

→ next page

Ex10 part 2

To Elaborate:

$$\begin{aligned} C_n &= \cancel{2C_{n-1}} + 1 \\ \cancel{2C_{n-1}} &= 2^2 \cancel{C_{n-2}} + 2 \\ \cancel{2^2 C_{n-2}} &= 2^3 C_{n-3} + 2^2 \end{aligned} \Bigg\} \text{Value of } C_n \text{ is a added up value of } 2(2^{n-x} - 2^x).$$

$\vdots$

Resulting the equation to equal $2^n - 1$.
But Note there is another Printout when $n == 0$. which adds another iteration (diverged and duplicated) of Printout.
Making the equation $2^{n+1} - 1$.
Note that inital value of C_1 is also replaced with the value of C_2 due to this extra iteration.

$$\boxed{\text{Closed form: } C_n = 2^{n+1} - 1 \text{ when } C_1 = 3}$$

Big O $= 2^{n+1} - 1 = 2^{n+1} = \underline{O(2^{n+1})}$